



Department of Computer Systems Engineering  
Mehran UET Jamshoro

# OBJECT ORIENTED PROGRAMMING

## PROJECT REPORT

### ATM PROGRAM IN JAVA

#### GROUP MEMBERS

**Abdul Rasheed | 20CS016**

**Hammas Khan | 20CS046**

**Vinita Chawla | 20CS012**

**Talhah Mirani | 20CS026**

**Abdul Samad | 20CS060**

SUBMITTED TO:

**Sir jamsher Bhambhro**

# About

This is a java based project that is simulating an ATM management system.

This project is completely based on Java and is coded within laws of object oriented programming.

## Features

- Login and Register
- Funds Information
- Saving Accounts
- Current Accounts
- Withdraw Funds
- Deposit Funds

## Conclusion

This project is a console based executable java class that is made on VS code editor. All the features mentioned above are working and fully tested.

The project is made with involvement of every group member.

# Code Snippets

```
import java.io.IOException;
import java.text.DecimalFormat;
import java.util.HashMap;
import java.util.InputMismatchException;
import java.util.Iterator;
import java.util.Map;
import java.util.Scanner;

public class OptionMenu {
    Scanner menuInput = new Scanner(System.in);
    DecimalFormat moneyFormat = new DecimalFormat("'$'###,##0.00");
    HashMap<Integer, Account> data = new HashMap<Integer, Account>();

    public void getLogin() throws IOException {
        boolean end = false;
        int customerNumber = 0;
        int pinNumber = 0;
        while (!end) {
            try {
                System.out.print("\nEnter your customer number: ");
                customerNumber = menuInput.nextInt();
                System.out.print("\nEnter your PIN number: ");
                pinNumber = menuInput.nextInt();
                Iterator it = data.entrySet().iterator();
                while (it.hasNext()) {
                    Map.Entry pair = (Map.Entry) it.next();
                    Account acc = (Account) pair.getValue();
                    if (data.containsKey(customerNumber) && pinNumber == acc.getPinNumber()) {
                        getAccountType(acc);
                        end = true;
                        break;
                    }
                }
            }
            if (!end) {
                System.out.println("\nWrong Customer Number or Pin Number");
            }
        } catch (InputMismatchException e) {
            System.out.println("\nInvalid Character(s). Only Numbers.");
        }
    }
}
```

```

public void getAccountType(Account acc) {
    boolean end = false;
    while (!end) {
        try {
            System.out.println("\nSelect the account you want to access: ");
            System.out.println(" Type 1 - Checkings Account");
            System.out.println(" Type 2 - Savings Account");
            System.out.println(" Type 3 - Exit");
            System.out.print("\nChoice: ");

            int selection = menuInput.nextInt();

            switch (selection) {
                case 1:
                    getChecking(acc);
                    break;
                case 2:
                    getSaving(acc);
                    break;
                case 3:
                    end = true;
                    break;
                default:
                    System.out.println("\nInvalid Choice.");
            }
        } catch (InputMismatchException e) {
            System.out.println("\nInvalid Choice.");
            menuInput.next();
        }
    }
}

public void getChecking(Account acc) {
    boolean end = false;
    while (!end) {
        try {
            System.out.println("\nCheckings Account: ");
            System.out.println(" Type 1 - View Balance");
            System.out.println(" Type 2 - Withdraw Funds");
            System.out.println(" Type 3 - Deposit Funds");
            System.out.println(" Type 4 - Transfer Funds");
            System.out.println(" Type 5 - Exit");
            System.out.print("\nChoice: ");

            int selection = menuInput.nextInt();

            switch (selection) {
                case 1:
                    System.out.println("\nCheckings Account Balance: " +
moneyFormat.format(acc.getCheckingBalance()));
                    break;
                case 2:
                    acc.getCheckingWithdrawInput();
                    break;
                case 3:
                    acc.getCheckingDepositInput();
                    break;
            }
        }
    }
}

```

```

        case 4:
            acc.getTransferInput("Checkings");
            break;
        case 5:
            end = true;
            break;
        default:
            System.out.println("\nInvalid Choice.");
    }
} catch (InputMismatchException e) {
    System.out.println("\nInvalid Choice.");
    menuInput.next();
}
}

}

public void getSaving(Account acc) {
    boolean end = false;
    while (!end) {
        try {
            System.out.println("\nSavings Account: ");
            System.out.println(" Type 1 - View Balance");
            System.out.println(" Type 2 - Withdraw Funds");
            System.out.println(" Type 3 - Deposit Funds");
            System.out.println(" Type 4 - Transfer Funds");
            System.out.println(" Type 5 - Exit");
            System.out.print("Choice: ");
            int selection = menuInput.nextInt();
            switch (selection) {
                case 1:
                    System.out.println("\nSavings Account Balance: " +
moneyFormat.format(acc.getSavingBalance()));
                    break;
                case 2:
                    acc.getSavingWithdrawInput();
                    break;
                case 3:
                    acc.getSavingDepositInput();
                    break;
                case 4:
                    acc.getTransferInput("Savings");
                    break;
                case 5:
                    end = true;
                    break;
                default:
                    System.out.println("\nInvalid Choice.");
            }
        } catch (InputMismatchException e) {
            System.out.println("\nInvalid Choice.");
            menuInput.next();
        }
    }
}
}
}

```

```

public void getSaving(Account acc) {
    boolean end = false;
    while (!end) {
        try {
            System.out.println("\nSavings Account: ");
            System.out.println(" Type 1 - View Balance");
            System.out.println(" Type 2 - Withdraw Funds");
            System.out.println(" Type 3 - Deposit Funds");
            System.out.println(" Type 4 - Transfer Funds");
            System.out.println(" Type 5 - Exit");
            System.out.print("Choice: ");
            int selection = menuInput.nextInt();
            switch (selection) {
                case 1:
                    System.out.println("\nSavings Account Balance: " +
moneyFormat.format(acc.getSavingBalance()));
                    break;
                case 2:
                    acc.getSavingWithdrawInput();
                    break;
                case 3:
                    acc.getSavingDepositInput();
                    break;
                case 4:
                    acc.getTransferInput("Savings");
                    break;
                case 5:
                    end = true;
                    break;
                default:
                    System.out.println("\nInvalid Choice.");
            }
        } catch (InputMismatchException e) {
            System.out.println("\nInvalid Choice.");
            menuInput.next();
        }
    }
}

public void createAccount() throws IOException {
    int cst_no = 0;
    boolean end = false;
    while (!end) {
        try {
            System.out.println("\nEnter your customer number ");
            cst_no = menuInput.nextInt();
            Iterator it = data.entrySet().iterator();
            while (it.hasNext()) {
                Map.Entry pair = (Map.Entry) it.next();
                if (!data.containsKey(cst_no)) {
                    end = true;
                }
            }
            if (!end) {
                System.out.println("\nThis customer number is already registered");
            }
        } catch (InputMismatchException e) {
            System.out.println("\nInvalid Choice.");
            menuInput.next();
        }
    }
    System.out.println("\nEnter PIN to be registered");
    int pin = menuInput.nextInt();
    data.put(cst_no, new Account(cst_no, pin));
    System.out.println("\nYour new account has been successfully registered!");
    System.out.println("\nRedirecting to login.....");
    getLogin();
}

```

```

public void mainMenu() throws IOException {
    data.put(952141, new Account(952141, 191904, 1000, 5000));
    data.put(123, new Account(123, 123, 20000, 50000));
    boolean end = false;
    while (!end) {
        try {
            System.out.println("\n Type 1 - Login");
            System.out.println(" Type 2 - Create Account");
            System.out.print("\nChoice: ");
            int choice = menuInput.nextInt();
            switch (choice) {
                case 1:
                    getLogin();
                    end = true;
                    break;
                case 2:
                    createAccount();
                    end = true;
                    break;
                default:
                    System.out.println("\nInvalid Choice.");
            }
        } catch (InputMismatchException e) {
            System.out.println("\nInvalid Choice.");
            menuInput.next();
        }
    }
    System.out.println("\nThank You for using this ATM.\n");
    menuInput.close();
    System.exit(0);
}
}

```

# Console Output

c:\Users\hamma\Downloads\Compressed\ATM-Machine-master - VS Code Console

Welcome to the ATM Project!

Type 1 - Login

Type 2 - Create Account

Choice:

c:\Users\hamma\Downloads\Compressed\ATM-Machine-master - VS Code Console

Enter your PIN number: 1234

Select the account you want to access:

Type 1 - Checkings Account

Type 2 - Savings Account

Type 3 - Exit

Choice: 1

Checkings Account:

Type 1 - View Balance

Type 2 - Withdraw Funds

Type 3 - Deposit Funds

Type 4 - Transfer Funds

Type 5 - Exit

Choice:

c:\Users\hamma\Downloads\Compressed\ATM-Machine-master - VS Code Console

Type 2 - Withdraw Funds

Type 3 - Deposit Funds

Type 4 - Transfer Funds

Type 5 - Exit

Choice: 3

Current Checkings Account Balance: \$0.00

Amount you want to deposit from Checkings Account: 45600

Current Checkings Account Balance: \$45,600.00

Checkings Account:

Type 1 - View Balance

Type 2 - Withdraw Funds

Type 3 - Deposit Funds

Type 4 - Transfer Funds

Type 5 - Exit

Choice: 1

Checkings Account Balance: \$45,600.00